

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM



ĐỒ ÁN CHUYÊN NGÀNH
THIẾT KẾ XE DÒ LINE VÀ TRÁNH VẬT CẢN

LỚP 06DHLTDT 2 KHÓA 2016-2018

GVHD: TRẦN HOÀN

SINH VIÊN THỰC HIỆN: LÂM PHÚ ĐỨC

TP.HỒ CHÍ MINH, 11/2017

TP. HỒ CHÍ MINH, tháng 11 năm 2017

TRƯỜNG ĐH CÔNG NGHIỆP THỰC

CỘNG HÒA XÃ HỘI CHỦ

PHẨM TP. HCM

NGHĨA VIỆT NAM

KHOA ĐIỆN – ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BỘ MÔN: ĐỒ ÁN CHUYÊN NGÀNH

TP.HCM, ngày.....tháng.....năm.....

**NHẬN XÉT ĐỒ ÁN CHUYÊN NGÀNH
CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tên đồ án:

Thiết kế xe dò line và tránh vật cản.

Sinh viên thực hiện:

Lâm Phú Đức

MSSV:2202162013

Giảng viên hướng dẫn:

Trần Hoàn

Đánh giá Luận văn

1. Về cuốn báo cáo:

Số trang _____ Số chương _____

Số bảng số liệu _____ Số hình vẽ _____

Số tài liệu tham khảo _____ Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

2. Về nội dung đồ án:

.....
.....
.....
.....

3. Về tính ứng dụng:

.....

.....

.....

.....

4. Về thái độ làm việc của sinh viên:

.....

.....

.....

.....

Đánh giá chung:

Điểm từng sinh viên:

(Họ tên sinh viên):...../10

Người nhận xét

(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

..... 000

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp.HCM, Ngày....Tháng.... Năm 2017

Giáo Viên Phản Biện

LỜI CẢM ƠN

Trong suốt quá trình thực hiện đồ án “Xe dò line và tránh vật cản”, em đã nhận được rất nhiều ý kiến đóng góp cũng như sự giúp đỡ và hướng dẫn tận tình của quý thầy cô và các bạn cùng lớp. Em xin chân thành cảm ơn các thầy cô, đặc biệt là thầy Trần Hoàn đã tạo mọi điều kiện thuận lợi giúp em hoàn thành xong đồ án này.

Do kiến thức còn hạn hẹp nên trong quá trình thực hiện đồ án em không thể tránh khỏi những sai sót, mong quý thầy cô trong hội đồng bảo vệ bỏ qua và có những góp ý để em có thể hoàn thiện đồ án của mình được tốt hơn nữa.

Em xin chân thành cảm ơn.

TP. Hồ Chí Minh, ngày 25 tháng 11 năm 2017

Sinh Viên Thực Hiện

Lâm Phú Đức

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT

TP. HCM

NAM

KHOA ĐIỆN – ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

BỘ MÔN: ĐỒ ÁN CHUYÊN NGÀNH

TP. HCM, ngày....tháng.....năm.....

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỒ ÁN: Thiết kế xe dò line và tránh vật cản.	
Giảng viên hướng dẫn: TS. Trần Hoàn	
Thời gian thực hiện: Từ ngày 17/9/2017 đến ngày 25/11/2017	
Sinh viên thực hiện: Lâm Phú Đức	
Nội dung đề tài: -Thiết kế xe dò line và tránh vật cản -Lập trình code sử dụng Arduino.	
Kế hoạch thực hiện: -Từ ngày 17/9/2017 đến ngày 20/9/2017: Nghiên cứu đề tài. -Từ ngày 21/9/2017 đến ngày 24/9/2017: Chuẩn bị linh kiện. -Từ ngày 25/9/2017 đến ngày 5/10/2017: Lắp ráp mô hình. -Từ ngày 6/10/2017 đến ngày 20/10/2017: Nghiên cứu lập trình Arduino. -Từ ngày 21/10/2017 đến ngày 5/11/2017: Viết code dò line. -Từ ngày 6/11/2017 đến ngày 16/11/2017: Viết code tránh vật cản. -Từ ngày 17/11/2017 đến ngày 25/11/2017: Viết báo cáo đồ án.	
Xác nhận của giảng viên hướng dẫn	TP. HCM, ngày 25 tháng 11 năm 2017 Sinh viên Lâm Phú Đức

MỤC LỤC

LỜI CẢM ƠN	4
DANH MỤC BẢNG BIỂU	8
DANH MỤC HÌNH ẢNH.....	8
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	9
1.1 Mục đích chọn đề tài.....	9
1.2 Mục tiêu đề tài... ..	9
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	10
2.1 Giới thiệu về Arduino... ..	10
2.2 Arduino Uno R3... ..	11
2.2.1 Cáp USB	11
2.2.2 IC Atmega 16U2... ..	11
2.2.3 Cổng nguồn ngoài.....	11
2.2.4 Cổng USB... ..	12
2.2.5 Nút Reset... ..	12
2.2.6 ICSP của Atmega 16U2.....	12
2.2.7 Chân xuất tín hiệu ra.....	12
2.2.8 IC Atmega 328.....	12
2.2.9 Chân ICSP của Atmega 328... ..	14
2.2.10 Chân lấy tín hiệu Analog... ..	14
2.2.11 Chân cấp nguồn cho cảm biến.....	15
2.2.12 Các linh kiện khác trên board Arduino.....	15
2.3 Linh kiện sử dụng	20
2.3.1 Động cơ và bánh xe	18
2.3.2 Cảm biến dò đường thanh 5 led.....	18

2.3.3 Mạch cầu H L298	19
2.3.4 Cảm biến siêu âm.....	23
2.3.4.1 Nguyên lý hoạt động của cảm biến siêu âm... ..	24
2.3.4.2 Cấu tạo cảm biến siêu âm	24
CHƯƠNG 3: CƠ SỞ THỰC HIỆN	25
3.1 Lắp ráp mô hình.....	25
3.2.1 Sơ đồ đi dây động cơ... ..	26
3.2.2 Sơ đồ đi dây cảm biến dò line... ..	27
3.2.3 Sơ đồ đi dây cảm biến khoảng cách... ..	27
3.2 Giải thuật điều khiển.....	28
3.2.1 Giải thuật dò line... ..	28
3.2.2 Giải thuật tránh vật cản.....	29
3.2.3 Lưu đồ giải thuật điều khiển.....	30
3.2.3.1 Chương trình chính.....	30
3.2.3.2 Trường hợp lệch trái	31
3.2.3.3 Trường hợp lệch phải	32
3.2.3.4 Tránh vật cản	33
CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM	34
4.1 Mô hình thực tế.....	34
4.2 Kết quả thực nghiệm.....	34
CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG ĐỀ TÀI	35
5.1 Kết quả	35
5.2 Hạn chế	35
5.3 Hướng phát triển của đề tài.....	35
PHỤ LỤC	36
1. Code chương trình	36
2. Giới thiệu phần mềm sử dụng... ..	45

3. Hướng dẫn cài đặt phần mềm.....	45
TÀI LIỆU THAM KHẢO	50
DANH MỤC BẢNG BIỂU	8
Bảng 1: Thông số mạch Arduino Uno R3	17
Bảng 2: Chức năng các chân L298	20
Bảng 3: Chế độ hoạt động mạch cầu H	21
DANH MỤC HÌNH ẢNH	8
Hình 2.1 Mạch Arduino Uno R3	11
Hình 2.2 Sơ đồ chân Atmega 328.....	13
Hình 2.3 Một số chức năng của các chân trên Arduino	17
Hình 2.4 Động cơ và bánh xe	18
Hình 2.5 Cảm biến dò đường.....	18
Hình 2.6 Sơ đồ nguyên lý mạch cầu H.....	21
Hình 2.7 Mạch cầu H L298	22
Hình 2.8 Cảm biến siêu âm.....	23
Hình 2.9 Sơ đồ chân cảm biến siêu âm	25
Hình 3.1 Sơ đồ nối dây động cơ	26
Hình 3.2 Sơ đồ nối dây cảm biến dò line	27
Hình 3.3 Sơ đồ nối dây cảm biến khoảng cách	27
Hình 4.1 Mô hình thực tế.....	34
Hình 1: Bảng thông báo bắt đầu cài đặt phần mềm.....	46
Hình 2: Thông báo điều khoản phần mềm	46
Hình 3: Các lựa chọn khi cài đặt	47
Hình 4: Chọn đường dẫn cho phần mềm.....	47
Hình 5: Quá trình cài đặt đang được thực hiện	48
Hình 6: Cài đặt driver USB cho IDE.....	48
Hình 7: Phần mềm đã được cài xong.....	49

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Mục đích chọn đề tài

Ngày nay, robotic đã đạt được những thành tựu to lớn trong sản xuất công nghiệp cũng như trong đời sống. Sản xuất robot là ngành công nghiệp trị giá hàng tỉ USD và ngày càng phát triển mạnh, trong các họ robot chúng ta không thể không nhắc tới mobile robot với những đặc thù riêng mà các loại robot khác không có. Mobile robot có thể di chuyển một cách rất linh hoạt, do đó tạo nên không gian hoạt động lớn và cho đến nay nó đã dần khẳng định vai trò quan trọng không thể thiếu trong nhiều lĩnh vực, thu hút được rất nhiều sự đầu tư và nghiên cứu. Mobile robot cũng được chia ra làm nhiều loại: robot học đường đi, robot dò đường line, robot tránh vật cản, robot tìm đường cho mê cung,...trong số đó robot dò đường line, tránh vật cản dễ dàng ứng dụng nhiều trong cuộc sống. Việc phát triển loại robot này sẽ phục vụ rất đắc lực cho con người

Robot dò line và tránh vật cản vừa có nhiều ứng dụng trong thực tế vừa dễ dàng để sinh viên vận dụng những kiến thức tiếp thu được trên giảng đường vào nó. Với những kết cấu cơ khí đơn giản nhưng lại có thể kết hợp được với khá nhiều thành phần điện tử (encoder, sensor xác định đường line, sensor đo khoảng cách...) nên những Robot này rất phù hợp để sinh viên học tập và nghiên cứu thêm về ngành Tự động hóa một cách cụ thể.

1.2. Mục tiêu đề tài:

- Thiết kế, xây dựng mô hình xe dò line và tránh vật cản.
- Nghiên cứu và lập trình code vi điều khiển sử dụng Arduino

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

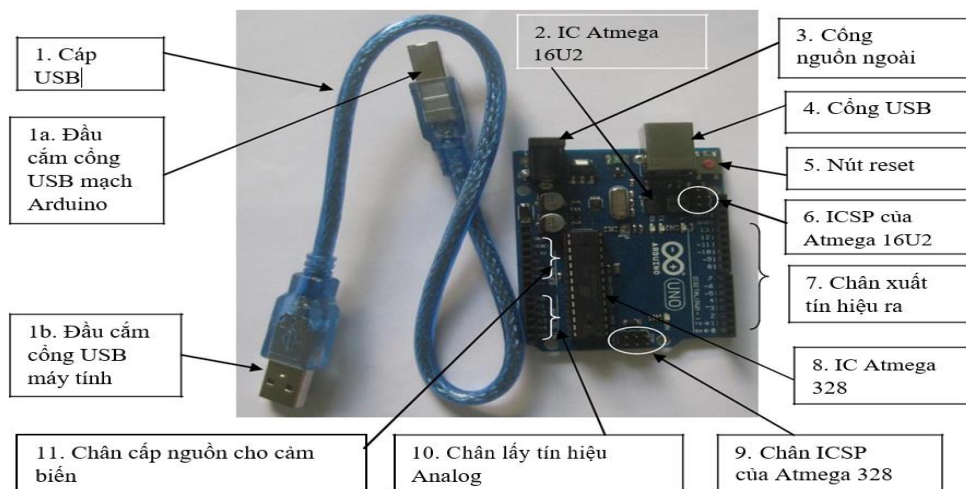
2.1. Giới thiệu về Arduino.

Arduino là một bo mạch vi điều khiển do một nhóm giáo sư và sinh viên Ý thiết kế và đưa ra đầu tiên vào năm 2005. Mạch Arduino được sử dụng để cảm nhận và điều khiển nhiều đối tượng khác nhau. Nó có thể thực hiện nhiều nhiệm vụ từ lấy tín hiệu từ cảm biến đến điều khiển đèn, động cơ, và nhiều đối tượng khác.

Ngoài ra mạch còn có khả năng liên kết với nhiều module khác nhau như module đọc thẻ từ, ethernet shield, sim900Ađể tăng khả năng ứng dụng của mạch.

Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM, Atmel 32-bit,.... Hiện phần cứng của Arduino có tất cả 6 phiên bản. Tuy nhiên phiên bản thường được sử dụng nhiều nhất là Arduino Uno và Arduino Mega. Arduino Uno được sử dụng rất rộng rãi trên thế giới, rất nhiều ví dụ trên youtube hoặc các trang hướng dẫn về Arduino sử dụng mạch này. Vì vậy đối với các bạn mới học Arduino, việc chọn Arduino Uno sẽ giúp các bạn có thể tự học dễ dàng.

2.2. Arduino Uno R3



Hình 2.1: Mạch Arduino Uno R3.

2.2.1. Cáp USB

Đây là dây cáp thường được bán kèm theo board, dây cáp dùng để cắm vào máy tính để nạp chương trình cho board và dây đồng thời cũng lấy nguồn từ nguồn usb của máy tính để cho board hoạt động. Ngoài ra cáp USB còn được dùng để truyền dữ liệu từ board Arduino lên máy tính. Dây cáp có 2 đầu, đầu 1a được dùng để cắm vào cổng USB trên board Arduino, đầu 1b dùng để cắm vào cổng USB trên máy tính.

2.2.2. IC Atmega 16U2

IC này được lập trình như một bộ chuyển đổi USB – Serial dùng để giao tiếp với máy tính thông qua giao thức Serial (dùng cổng COM).

2.2.3. Cổng nguồn ngoài

Cổng nguồn ngoài nhằm sử dụng nguồn điện bên ngoài như pin, bình acquy hay các adapter cho board Arduino hoạt động. Nguồn điện cấp vào cổng này là nguồn DC có hiệu điện thế từ 6V đến 20V, tuy nhiên hiệu điện thế tốt nhất mà nhà sản xuất khuyến dùng là từ 7 đến 12V.

2.2.4. Cổng USB

Cổng USB trên board Arduino dùng để kết nối với cáp USB.

2.2.5. Nút reset

Nút reset được sử dụng để reset lại chương trình đang chạy. Đôi khi chương trình chạy gặp lỗi, người dùng có thể reset lại chương trình.

2.2.6. ICSP của ATmega 16U2

ICSP là chữ viết tắt của In-Circuit Serial Programming. Đây là các chân giao tiếp SPI của chip Atmega 16U2. Các chân này thường ít được sử dụng trong các dự án về Arduino.

2.2.7. Chân xuất tín hiệu ra

Có tất cả 14 chân xuất tín hiệu ra trong Arduino Uno, những chân có dấu ~ là những chân có thể băm xung (PWM), tức có thể điều khiển tốc độ động cơ hoặc độ sáng của đèn.

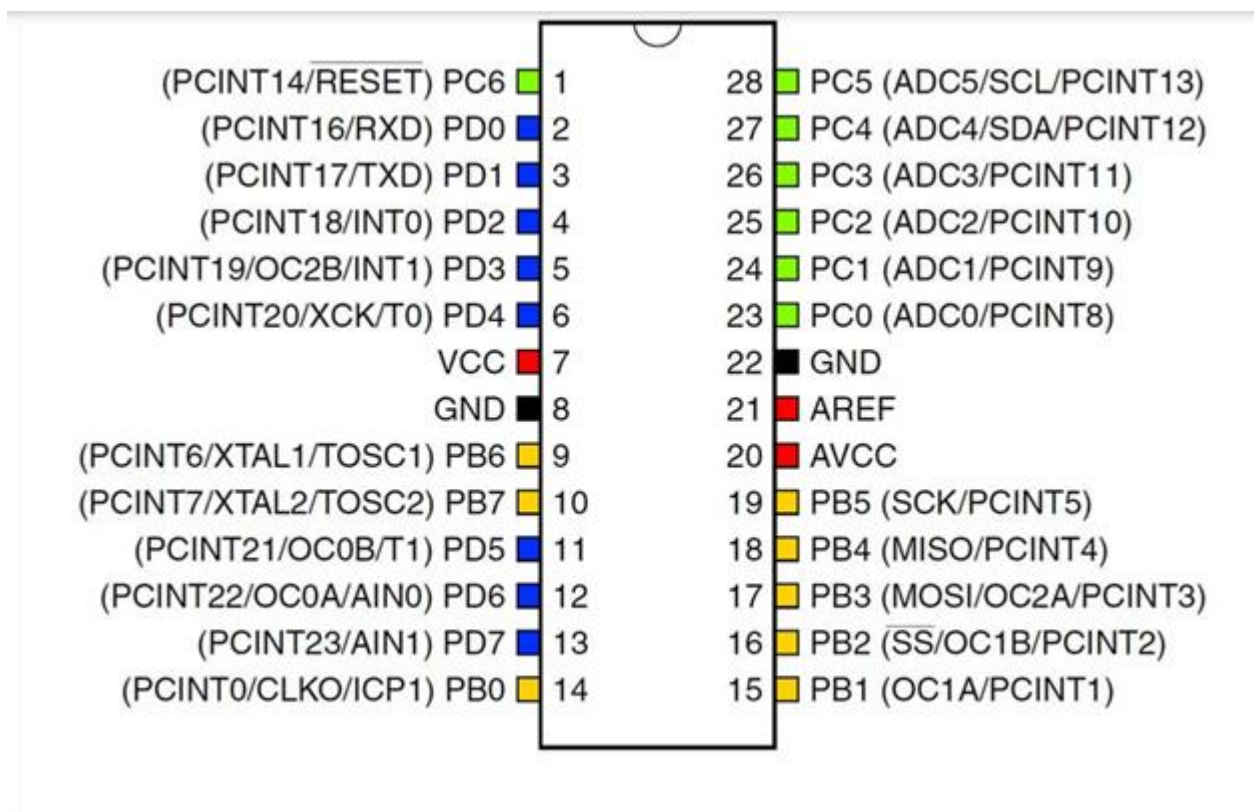
2.2.8. IC ATmega 328

IC Atmega 328 là linh hồn của board mạch Arduino Uno, IC này được sử dụng trong việc thu thập dữ liệu từ cảm biến, xử lý dữ liệu, xuất tín hiệu ra.

Atmega328 là một chip vi điều khiển được sản xuất bởi hãng Atmel thuộc họ MegaAVR có sức mạnh hơn hẳn Atmega8. Atmega 328 là một bộ vi điều khiển 8

bít dựa trên kiến trúc RISC bộ nhớ chương trình 32KB ISP flash có thể ghi xóa hàng nghìn lần, 1KB EEPROM, một bộ nhớ RAM vô cùng lớn trong thế giới vi xử lý 8 bit (2KB SRAM)

Với 23 chân có thể sử dụng cho các kết nối vào hoặc ra i/O, 32 thanh ghi, 3 bộ timer/counter có thể lập trình, có các ngắt nội và ngoại (2 lệnh trên một vector ngắt), giao thức truyền thông nối tiếp USART, SPI, I2C. Ngoài ra có thể sử dụng bộ biến đổi số tương tự 10 bit (ADC/DAC) mở rộng tới 8 kênh, khả năng lập trình được watchdog timer, hoạt động với 5 chế độ nguồn, có thể sử dụng tới 6 kênh điều chế độ rộng xung (PWM), hỗ trợ bootloader.



Hình 2.2: Sơ đồ chân Atmega328.

Atmega328 có khả năng hoạt động trong một dải điện áp rộng (1.8V – 5.5V), tốc độ thực thi (thông lượng) 1MIPS trên 1MHz

Ngày nay vi điều khiển Atmega328 thực sự được sử dụng phổ biến từ các dự án nhỏ của sinh viên, học sinh với giá thành rẻ, xử lý mạnh mẽ, tiêu tốn ít năng lượng (chế độ hoạt động: 0.2 mA, chế độ ngủ: 0.1 μ A, chế độ tích kiệm: 0.75 μ A) và sự hỗ trợ nhiệt tình của cộng đồng người dùng AVR. Và không thể không nhắc tới sự thành công của Vi điều khiển Atmega328 trong dự án mã nguồn mở Arduino với các module Arduino Uno (R3), Arduino Nano, Arduino Pro mini những sản phẩm dẫn dắt chúng ta vào thế giới mã nguồn mở để hoàn thành một chương trình trong “nháy mắt”.

Thông số chính Atmega328P-PU:

- + Kiến trúc: AVR 8bit
- + Xung nhịp lớn nhất: 20 MHz
- + Bộ nhớ chương trình (FLASH): 32KB
- + Bộ nhớ EEPROM: 1KB
- + Bộ nhớ RAM: 2KB
- + Điện áp hoạt động rộng: 1.8V – 5.5V
- + Số timer: 3 timer gồm 2 timer 8-bit và 1 timer 16-bit
- + Số kênh xung PWM: 6 kênh (1timer 2 kênh)

2.2.9Chân ICSP của ATmega 328

Các chân ICSP của ATmega 328 được sử dụng cho các giao tiếp SPI (Serial Peripheral Interface), một số ứng dụng của Arduino có sử dụng chân này, ví dụ như sử dụng module RFID RC522 với Arduino hay Ethernet Shield với Arduino

2.2.10. Chân lấy tín hiệu Analog

Các chân này lấy tín hiệu Analog (tín hiệu tương tự) từ cảm biến để IC Atmega 328 xử lý. Có tất cả 6 chân lấy tín hiệu Analog, từ A0 đến A5.

2.2.11. Chân cấp nguồn cho cảm biến

Các chân này dùng để cấp nguồn cho các thiết bị bên ngoài như role, cảm biến, RC servo,...trên khu vực này có sẵn các chân GND (chân nối đất, chân âm), chân 5V, chân 3.3V như được thể hiện ở hình 2. Nhờ những chân này mà người sử dụng không cần thiết bị biến đổi điện khi cấp nguồn cho cảm biến, role, RC servo...Ngoài ra trên khu vực này còn có chân Vin và chân reset, chân IOREF. Tuy nhiên các chân này thường ít được sử dụng nên trong tài liệu này xin không đi sâu về nó.

2.2.12. Các linh kiện khác trên board Arduino Uno

Ngoài các linh kiện đã liệt kê bên trên, Arduino Uno còn 1 số linh kiện đáng chú ý khác. Trên bo có tất cả 4 đèn led, bao gồm 1 led nguồn (led ON nhằm cho biết bo đã được cấp nguồn), 2 led Tx và Rx, 1 led L. Các led Tx và Rx sẽ nhấp nháy khi có dữ liệu truyền từ board lên máy tính hoặc ngược lại thông qua cổng USB. Led L được kết nối với chân số 13. Led này được gọi là led on board (tức led trên bo), led này giúp người dùng có thể thực hành các bài đơn giản mà không cần dùng thêm led ngoài.

Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân digital có các chức năng đặc biệt như sau:

- 2 chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy là kết nối Serial không dây.

Nếu không cần giao tiếp serial, bạn không nên sử dụng 2 chân này nếu không cần thiết.

- Phân giải 8bit (giá trị từ 0 \rightarrow 255 tương ứng với 0V \rightarrow 5V) bằng hàm Chân PWM (~): 3, 5, 6, 9, 10, và 11: cho phép bạn xuất ra xung PWM với độ analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

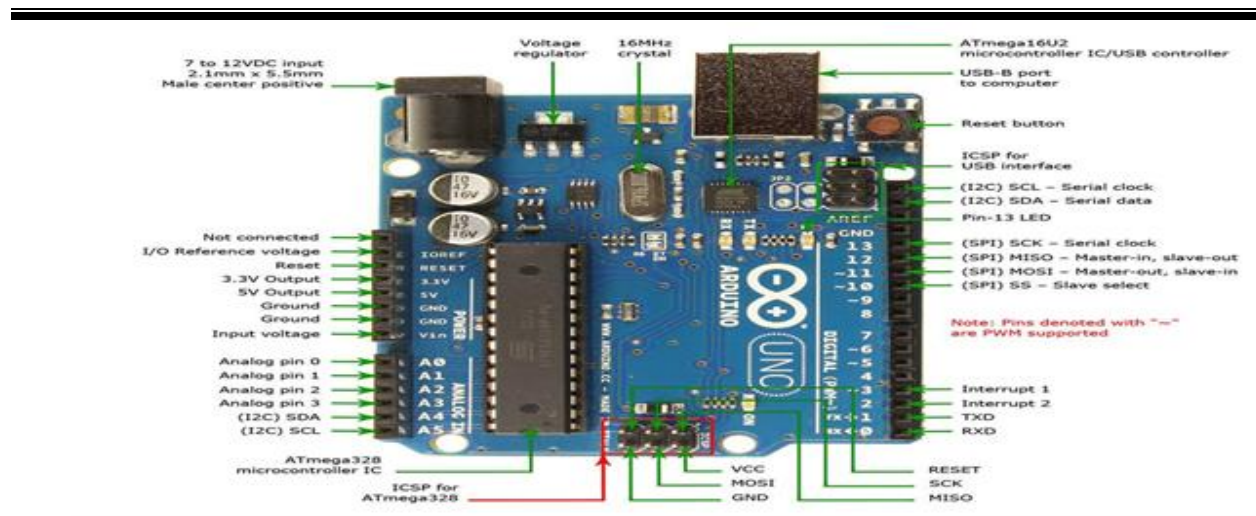
- Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

- LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

- Arduino UNO có 6 chân analog (A0 \rightarrow A5) cung cấp độ phân giải tín hiệu 10bit (0 \rightarrow 1023) để đọc giá trị điện áp trong khoảng 0V \rightarrow 5V. Với chân AREF trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V \rightarrow 2.5V với độ phân giải vẫn là 10bit. Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

Trong 14 chân ra của bo còn có 2 chân 0 và 1 có thể truyền nhận dữ liệu nối tiếp TTL. Có một số ứng dụng cần dùng đến tính năng này, ví dụ như ứng dụng điều khiển mạch Arduino Uno qua điện thoại sử dụng bluetooth HC05.

Thêm vào đó, chân 2 và chân 3 cũng được sử dụng cho lập trình ngắt (interrupt), đồng thời còn 1 vài chân khác có thể được sử dụng cho các chức năng khác, như được thể hiện ở hình 3. Bảng 1 thể hiện thêm các thông số cho bo Arduino Uno R3.



Hình 2.3: Một số chức năng của các chân trên Arduino.

Thông số board Arduino Uno R3

Vi điều khiển	ATmega328 (họ 8bit)
Điện áp hoạt động	5V – DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	30mA
Điện áp vào khuyến dùng	7-12V – DC
Điện áp vào giới hạn	6-20V – DC
Số chân Digital I/O	14 (6 chân PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA

Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Bảng 2: Thông số mạch Arduino Uno R3

2.3. Linh kiện sử dụng

2.3.1. Động cơ (Motor) + Bánh Xe

Điện áp hoạt động: 3-12VDC

Dòng điện tiêu thụ: 110-200mA

Tỉ số truyền: 1:48

Tốc độ quay:

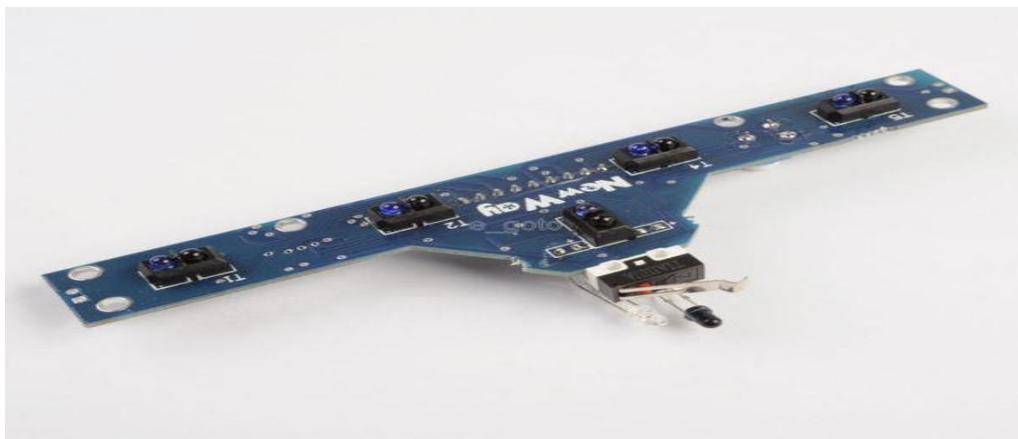
3V ~ 125 rpm

5V ~ 208 rpm



Hình 2.4: Động cơ và bánh xe.

2.3.2. Cảm biến dò đường thanh 5 led (Line Sensor)



Hình 2.5: Cảm biến dò đường.

- Phát hiện vùng màu đen và vùng màu sáng
- Hoạt động dựa trên nguyên lý của tia hồng ngoại bị hấp thu khi chiếu vào bề mặt màu đen và phản xạ với các bề mặt màu sáng.
- 5 cảm biến hồng ngoại hướng xuống đất: cảm biến nào nằm trên vùng màu đen sẽ cho giá trị 0, ngược lại là 1.
- Điện áp hoạt động: 3.3 ~ 5VDC
- Có LED hiển thị ngõ ra cho từng cảm biến
- Tích hợp 5 cảm biến dò line, 1 cảm biến tránh vật cản và 1 công tắc hành trình báo chạm vật.
- Ngõ ra gồm 7 chân tín hiệu của cảm biến dạng số và 2 ngõ vào cấp nguồn cho thiết bị.
- Khoảng cách phát hiện 1cm~2.5cm
- Kích thước: 128 x 45 x 12mm

2.3.3. Mạch Cầu H L298

Để điều khiển động cơ DC ta sử dụng L298. IC L298 là mạch tích hợp đơn chip có kiểu vỏ công suất 15 chân (multiwatt 15) hoặc PowerSO20 (linh kiện dán công suất). Là IC mạch cầu đôi (dual full-bridge) có khả năng hoạt động ở điện thế cao, dòng cao. Nó được thiết kế tương thích chuẩn TTL và lái tải cảm kháng như relay, cuộn solenoid, động cơ DC và động cơ bước. Nó có 2 chân enable để cho phép/không cho phép IC hoạt động, độc lập với các chân tín hiệu vào. Cực phát (emitter) của transistor dưới của mỗi mạch cầu được nối với nhau và nối ra chân ngoài để nối với điện trở cảm ứng dòng khi cần.

Tần công suất ngõ ra: IC L298 tích hợp 2 tầng công suất (A, B). Tần công suất chính là mạch cầu và ngõ ra của nó có thể lái các loại tải cảm thông dụng ở nhiều chế độ hoạt động khác nhau (tùy thuộc vào sự điều khiển ở ngõ vào) Dòng điện từ chân ngõ ra chảy qua tải đến chân cảm ứng dòng: điện trở ngoài RSA, RSB cho phép việc cảm ứng cường độ dòng điện này.

Tần ngõ vào: Mỗi cầu được điều khiển bởi 3 cổng ngõ vào In1, In2, EnA (và In3, In4, EnB cho cầu còn lại). Các chân In có tác dụng khi chân En ở mức cao, khi chân En ở mức thấp, các chân ngõ vào In ở trạng thái cấm. Tất cả các chân đều tương thích với chuẩn TTL.

Chức năng của các chân:

MW.15	PowerSO	Tên	Chức năng
1;15	2;19	Sense A; Sense B	Nối chân này qua điện trở cảm ứng dòng xuống GND để điều khiển dòng tải
2;3	4;5	Out 1; Out 2	Ngõ ra của cầu A. Dòng của tải mắc giữa hai chân này được qui định bởi chân 1.
4	6	VS	Chân cấp nguồn cho tầng công suất. Cần có một tụ điện không cảm kháng 100nF nối giữa chân này và chân GND
5;7	7;9	Input 1; Input 2	Chân ngõ vào của cầu A, tương thích chuẩn TTL
6;11	8;14	Enable A; EnableB	Chân ngõ vào enable (cho phép) tương thích chuẩn TTL. Mức thấp ở chân này sẽ cấm (disable) ngõ ra cầu A (đối với chân EnableA) và/hoặc cầu B (đối với chân EnableB)
8	1,10,11,20	GND	Chân đất (Ground)
9	12	VSS	Chân cấp nguồn cho khối logic. Cần có tụ điện 100nF nối giữa chân này với GND
10; 12	13;15	Input 3; Input 4	Các chân logic ngõ vào của cầu B
13; 14	16;17	Out 3; Out 4	Ngõ ra của cầu B. Dòng của tải mắc giữa hai chân này được qui định bởi chân 15.
-	3;18	N.C.	Không kết nối (bỏ trống)

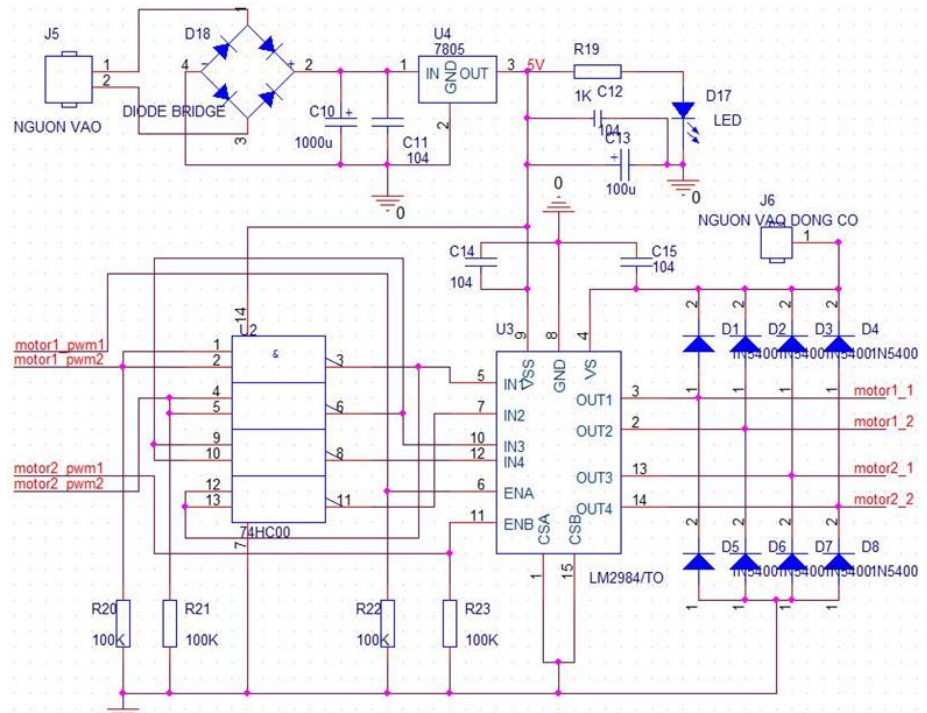
Bảng 3: Chức năng các chân L298

Chế độ hoạt động:

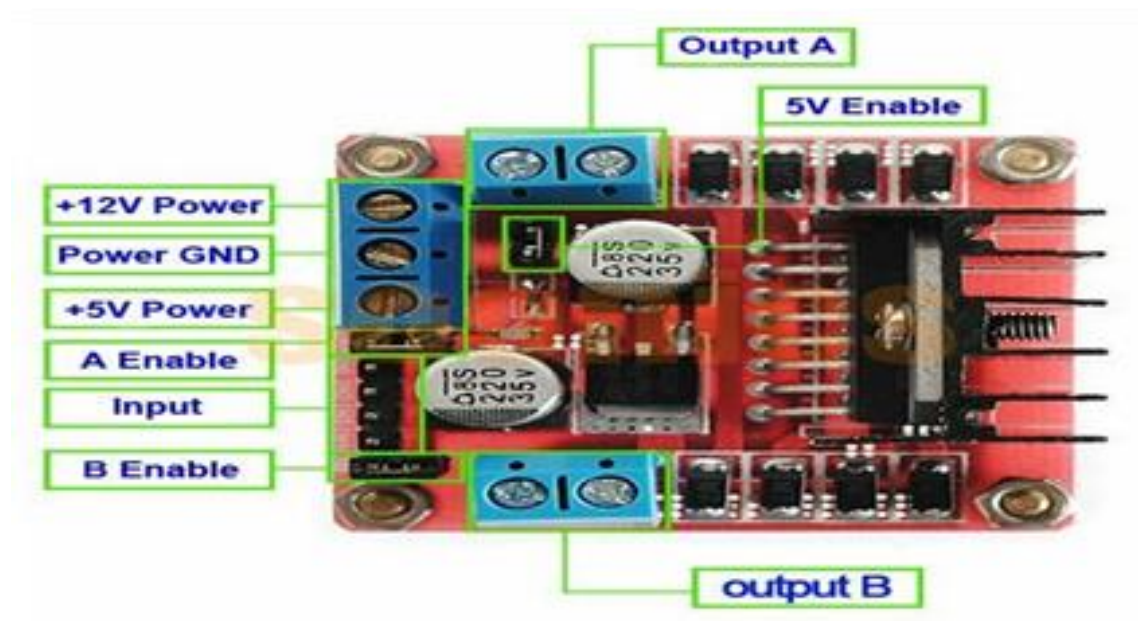
En	In1	In2	Chế độ hoạt động
0	x	x	Không hoạt động
1	0	0	Phanh
1	0	1	Động cơ quay trái (phải)
1	1	0	Động cơ quay phải (trái)
1	1	1	Phanh

Bảng 4: Chế độ hoạt động mạch cầu H.

Sơ đồ nguyên lý:



Hình 2.6: Sơ đồ nguyên lí mạch cầu H L298



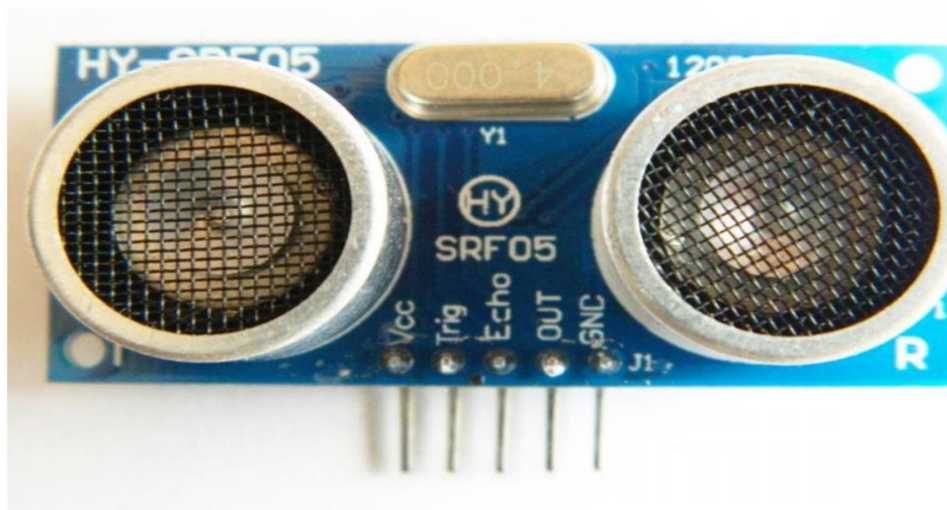
Hình 2.7: Mạch cầu H L298

Thông số kỹ thuật:

- Driver: L298N tích hợp hai mạch cầu H.
- Điện áp điều khiển: +5 V ~ +12 V
- Dòng tối đa cho mỗi cầu H là: 2A (\Rightarrow 2A cho mỗi motor)
- Điện áp của tín hiệu điều khiển: +5 V ~ +7 V
- Dòng của tín hiệu điều khiển: 0 ~ 36mA (Arduino có thể đến 40mA)
- Công suất hao phí: 20W (khi nhiệt độ $T = 75\text{ }^{\circ}\text{C}$)
- Nhiệt độ bảo quản: $-25\text{ }^{\circ}\text{C} \sim +130\text{ }^{\circ}\text{C}$

2.3.4. Cảm biến siêu âm

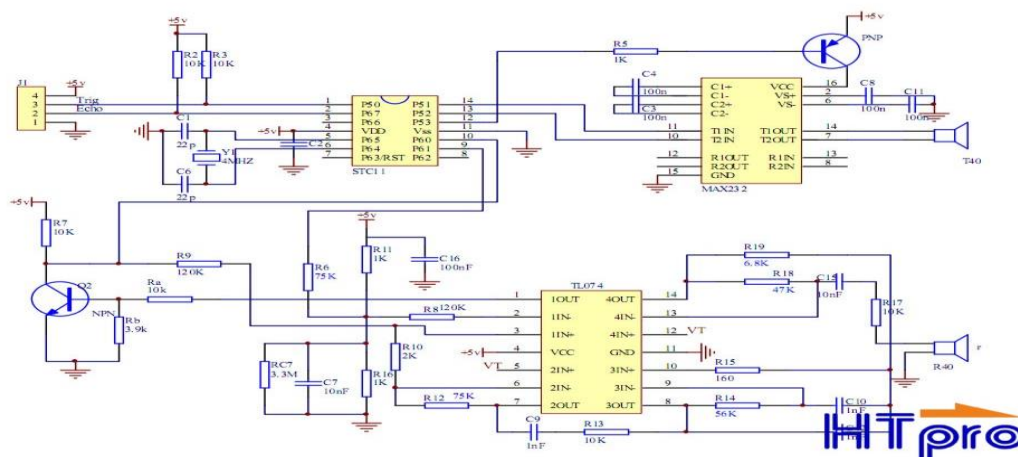
2.3.4.1 Nguyên lý hoạt động của cảm biến siêu âm



Hình 2.8: Cảm biến siêu âm

Cảm biến siêu âm là một trong số những cảm biến được ứng dụng nhiều nhất, trong các lĩnh vực thông thường như robot, mô hình hay đến những lĩnh vực chuyên dụng như đo độ sâu, quét địa hình đáy biển, đo độ dày bê tông trong xây dựng hoặc các ứng dụng trong quân sự,... Sau đây, mình sẽ phân tích cấu tạo và nguyên lý hoạt động của một trong số những cảm biến này, đó là HC-SR05.

Sơ đồ nguyên lý cảm biến siêu âm



Hình 2.9: Sơ đồ chân cảm biến siêu âm

2.3.4.2. Cấu tạo của cảm biến siêu âm

Cấu tạo gồm 3 phần:

Hiệu Phát tín

Các đầu phát và đầu thu siêu âm là các loa gồm được chế tạo đặc biệt, hoạt động phát siêu âm có cường độ cao nhất ở một tần số nào đó (thường là 40kHz cho các ứng dụng đo khoảng cách). Các loa này cần có nguồn tín hiệu điều khiển có điện áp cao mới phát tốt được (theo datasheet thì là $\sim 30V$). Chính vì vậy trong phần phát, phần đệm công suất sử dụng một con MAX232 làm nhiệm vụ đệm. Nó sẽ lấy tín hiệu từ bộ điều khiển, khuếch đại biên độ lên $\pm 30V$ cung cấp cho loa gồm.

Để tiết kiệm nguồn cho module cảm biến, phần cấp điện cho MAX232 được điều khiển.

Thu tín hiệu

Khi loa gồm làm đầu thu (loa này được chế tạo chỉ nhạy với một tần số nào đó-40KHz) thu được sóng siêu âm, nó sẽ phát ra một điện thế giữa hai cực. Điện thế

này là rất nhỏ, vì vậy nó được đưa qua một OPAM, ở đây là TL072 (Một số module sử dụng LM324,...). Tín hiệu này liên tục được khuếch đại biên độ và cuối cùng là đưa qua một bộ so sánh, kết hợp với tín hiệu từ bộ điều khiển để đưa về bộ điều khiển thông qua một trạm NPN

Phần xử lý và điều khiển

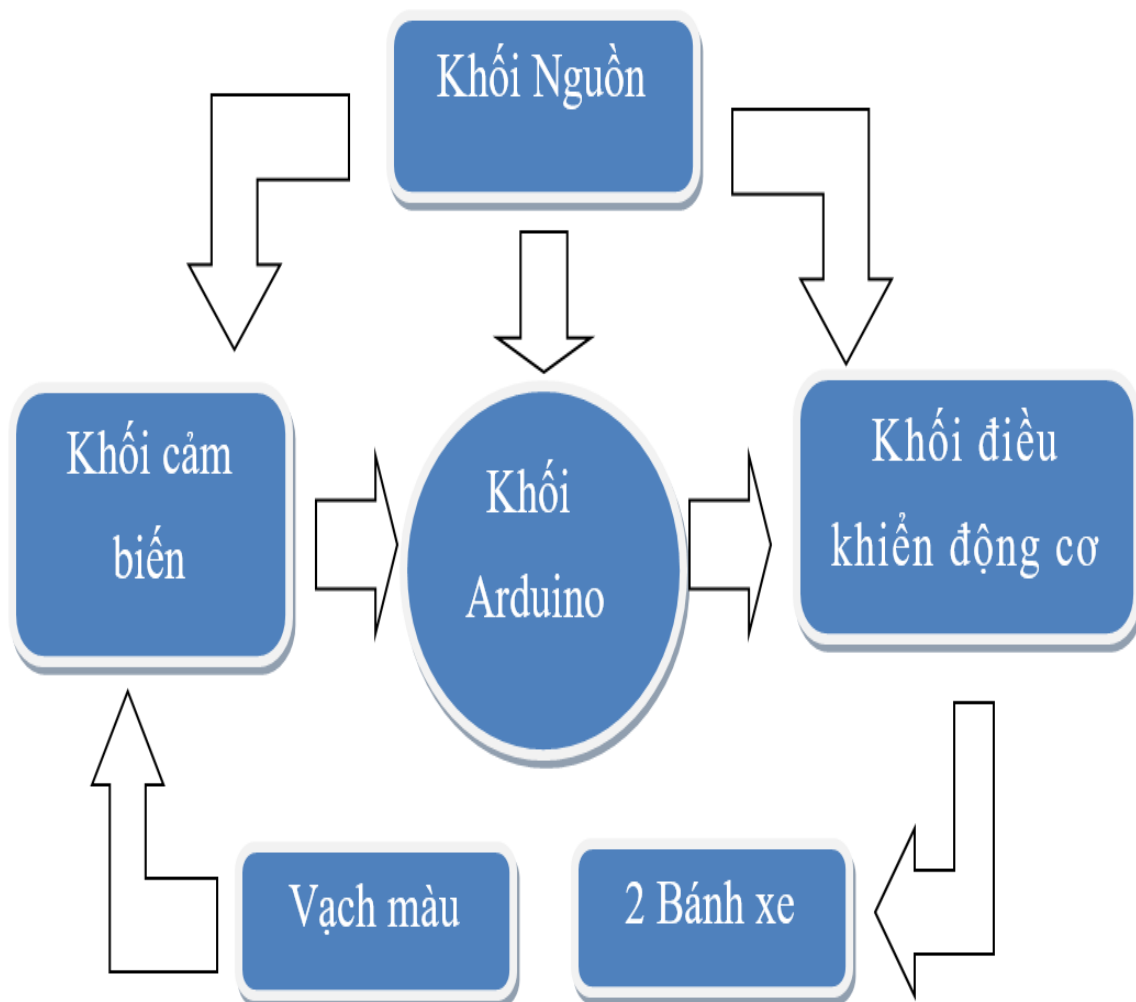
Phần xử lý, điều khiển thường sử dụng một vi điều khiển (PIC16F688, STC11,...) làm nhiệm vụ phát xung, xử lý tính toán thời gian từ khi phát đến khi thu được sóng siêu âm do nó phát ra nếu nhận được tín hiệu TRIG. Đến đây thì nguyên lý hoạt động thông thường của cảm biến này thì ai cũng biết rồi nhé (cấp xung TRIG, chờ đo độ rộng xung ECHO để tính toán thời gian....)

Tập trung lại, với những điều mình trình bày trên, các bạn có thể tự chế cảm biến siêu âm, làm module siêu âm đo khoảng cách lớn hơn hay như mình là chế đầu dò siêu âm đo khoảng cách dưới nước, dùng để đo độ sâu sông hồ.

CHƯƠNG 3: CƠ SỞ THỰC HIỆN

3.1. Lắp ráp mô hình

Sơ đồ khối các module trong mạch:

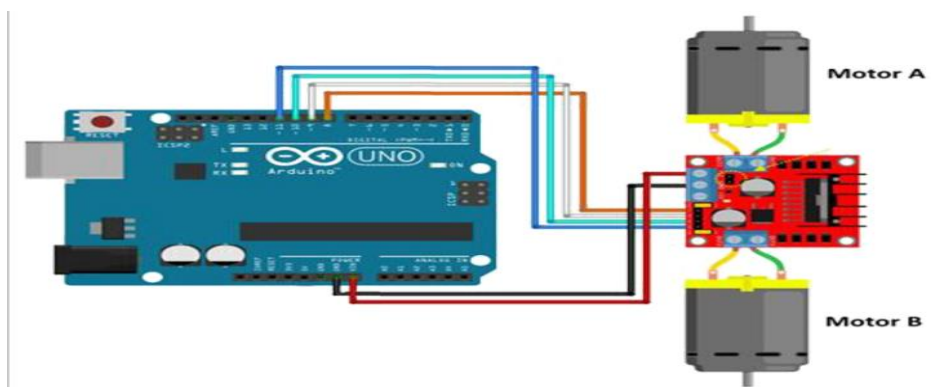


3.1.1. Sơ đồ đi dây động cơ:

Động cơ	Mạch khuếch đại
Động cơ trái	Out1
	Out2
Động cơ phải	Out3

Out4

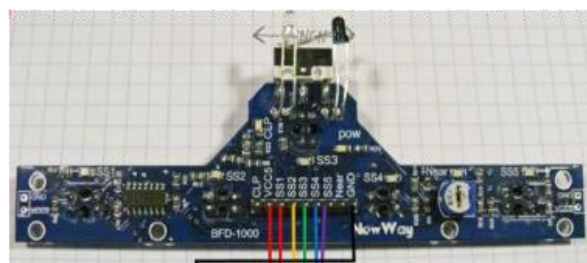
Mạch khuếch đại	Mạch arduino
In1	6
In2	7
In3	8
In4	9
+12V	Vin
GND	GND



Hình 3.1: Sơ đồ nối dây động cơ.

3.1.2. Sơ đồ đi dây cảm biến dò line

Cảm biến dò line	Mạch arduino
SS1	A0
SS2	A1
SS3	A2

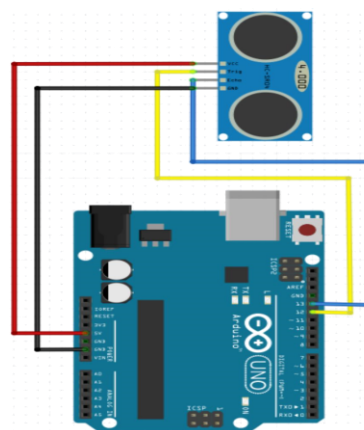


SS4	A3
SS5	A4
VCC	5V
GND	GND

Hình 3.2: Sơ đồ nối dây cảm biến dò line

3.1.3. Sơ đồ đi dây cảm biến khoảng cách

Cảm biến khoảng cách Mạch arduino	
VCC	5V
Trig	3
Echo	4
GND	GND



Hình 3.3: Sơ đồ đi dây cảm biến khoảng cách.

3.2. Giải thuật điều khiển

3.2.1. Giải thuật dò line:

Cảm biến dò line gồm 5 cảm biến hồng ngoại hướng xuống đất: cảm biến nào nằm trên vùng màu đen sẽ cho giá trị 0, ngược lại là 1. Vì vậy ta có các trường hợp:

Led 1	Led 2	Led 3	Led 4	Led 5		
0	0	0	0	0	Mất line trái	TH1
1	0	0	0	0	Lệch line trái mức 4	TH2
1	1	0	0	0	Lệch line trái mức 3	TH3
0	1	0	0	0	Lệch line trái mức 2	TH4
0	1	1	0	0	Lệch line trái mức 1	TH5
0	0	1	0	0	Giữa line	TH6
0	0	1	1	0	Lệch line phải mức 1	TH7
0	0	0	1	0	Lệch line phải mức 2	TH8
0	0	0	1	1	Lệch line phải mức 3	TH9
0	0	0	0	1	Lệch line phải mức 4	TH10
0	0	0	0	0	Mất line phải	TH11

Trong mỗi trường hợp, ta tiếp tục chia thành 3 trường hợp nhỏ hơn, đó là trạng thái trước đó (TTTĐ) rơi vào trường hợp nào (bằng, lớn hơn hay nhỏ hơn). Với mỗi trường hợp như vậy ta sẽ đặt giá trị tốc độ quay cho mỗi động cơ là khác nhau.

Trạng thái hiện tại là TH6:

- Nếu TTTĐ=TH6 thì ta giữ nguyên tốc độ của 2 động cơ (hoặc có thể cùng tăng giảm một lượng bằng nhau) để đảm bảo xe chạy thẳng.
- Nếu TTTĐ<6, xe đang có xu hướng rẽ sang trái và mong muốn đi thẳng nên ta cần rẽ phải một ít (tăng một lượng nhỏ tốc độ động cơ bên trái hoặc giảm một lượng nhỏ tốc độ động cơ bên phải)
- Nếu TTTĐ>6, xe đang có xu hướng rẽ sang phải và mong muốn đi thẳng nên ta cần rẽ trái một ít (tăng một lượng nhỏ tốc độ động cơ bên phải hoặc giảm một lượng nhỏ tốc độ động cơ bên trái).

Trạng thái hiện tại là TH2:

- Nếu $TTTĐ = TH2$, xe đang có xu hướng chạy khỏi line và mong muốn rẽ phải nên ta cần tăng trái hoặc giảm phải một lượng lớn.

- Nếu $TTTĐ < TH2$, xe đang mất phương hướng và mong muốn rẽ phải nên ta cần tăng trái hoặc giảm phải một lượng khá lớn.

Trạng thái hiện tại là TH10:

- Nếu $TTTĐ = TH10$, xe đang có xu hướng chạy khỏi line và mong muốn rẽ trái nên ta cần tăng phải hoặc giảm trái một lượng lớn.

- Nếu $TTTĐ > TH10$, xe đang mất phương hướng và mong muốn rẽ trái nên ta cần tăng phải hoặc giảm trái một lượng khá lớn.

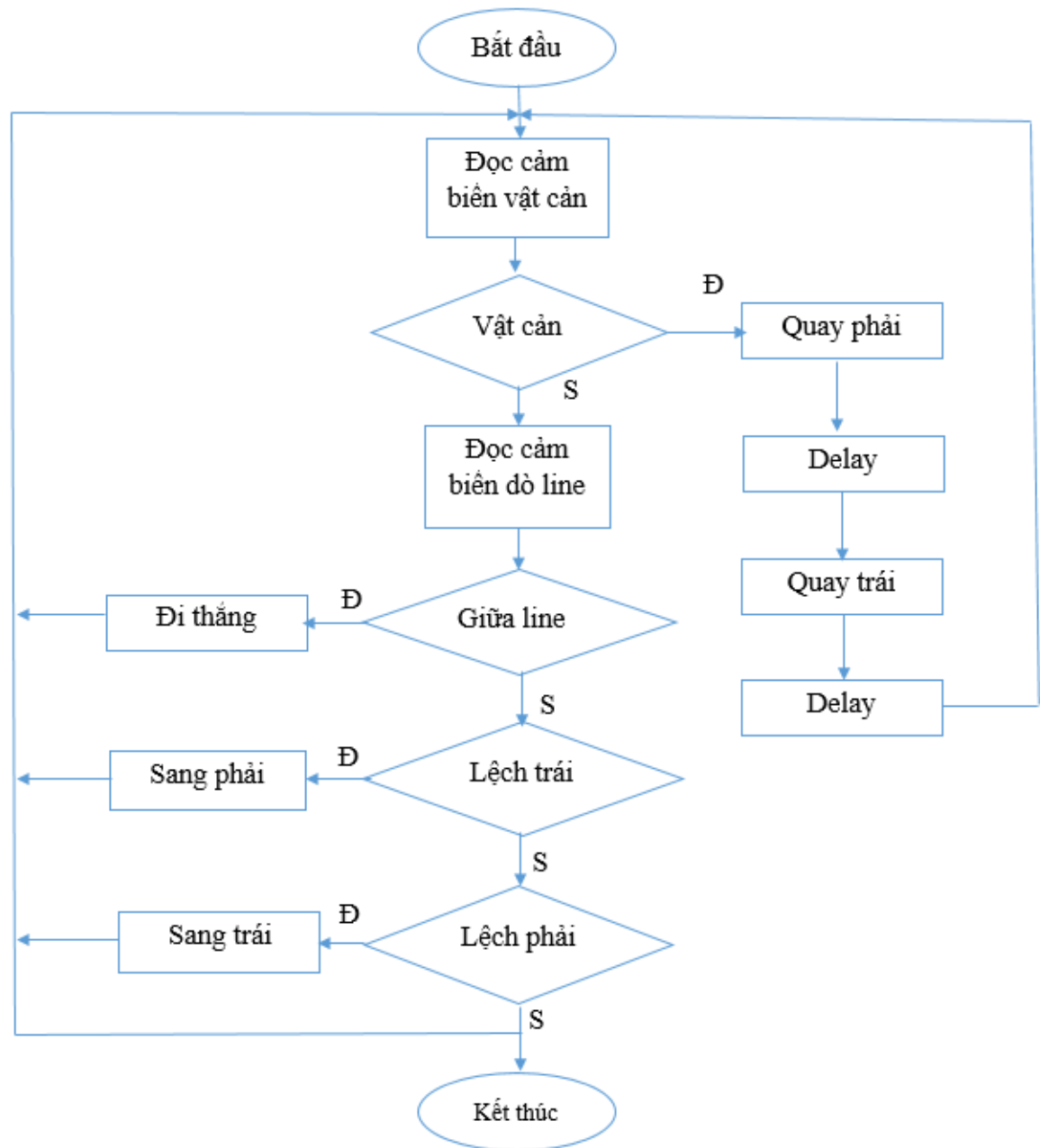
Quá trình này cần thực hiện nhiều lần để chọn ra những thông số phù hợp.

3.2.2. Giải thuật tránh vật cản

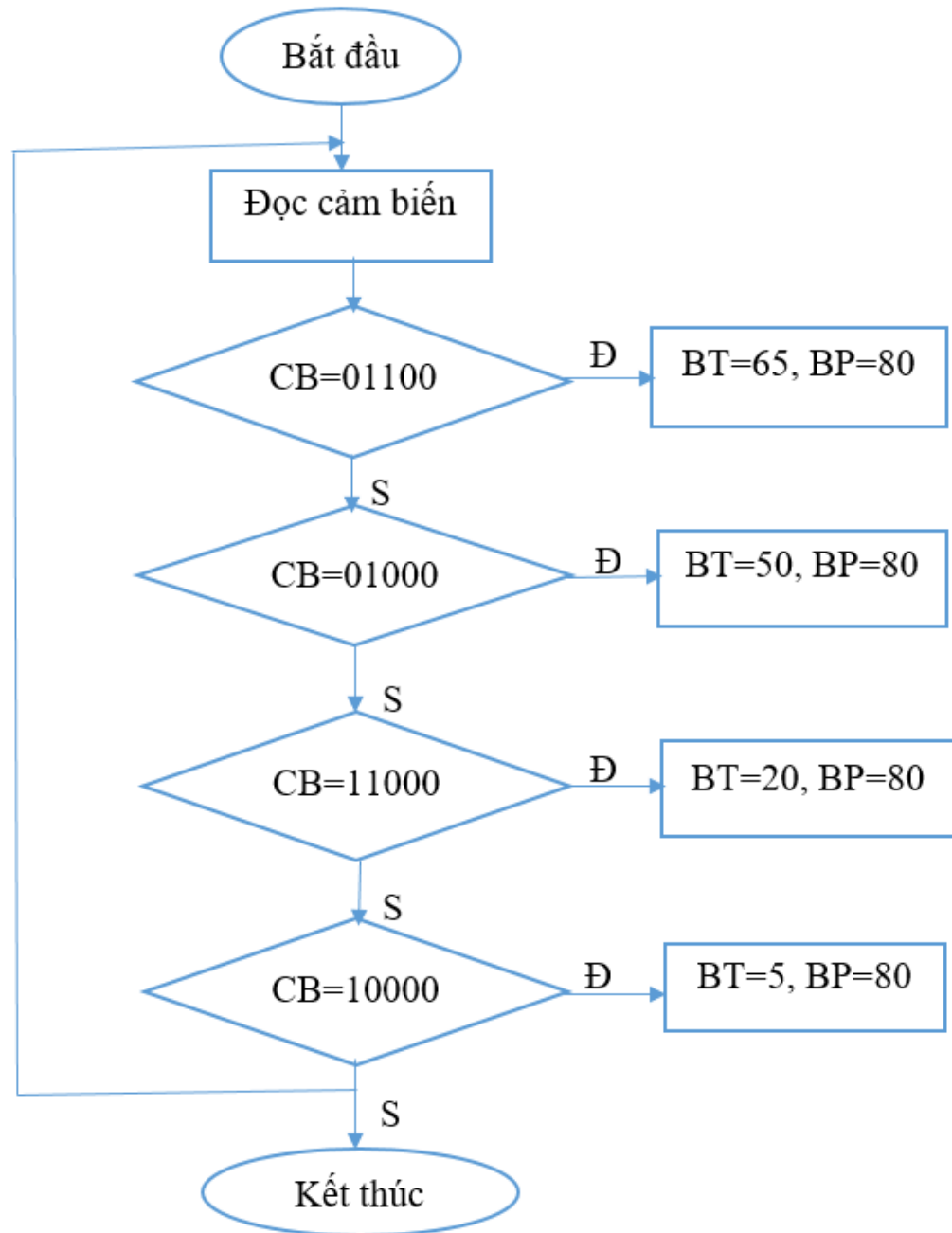
Xe sử dụng một cảm biến siêu âm HC-SR05 để đo khoảng cách giữa xe và vật cản. Nếu khoảng cách lớn hơn 20cm thì xe chạy thẳng. Nếu khoảng cách nhỏ hơn hoặc bằng 20cm cho xe dừng lại sau đó cho xe rẽ sang phải rồi rẽ sang trái để tiếp tục dò line.

3.2.3. Lưu đồ giải thuật điều khiển

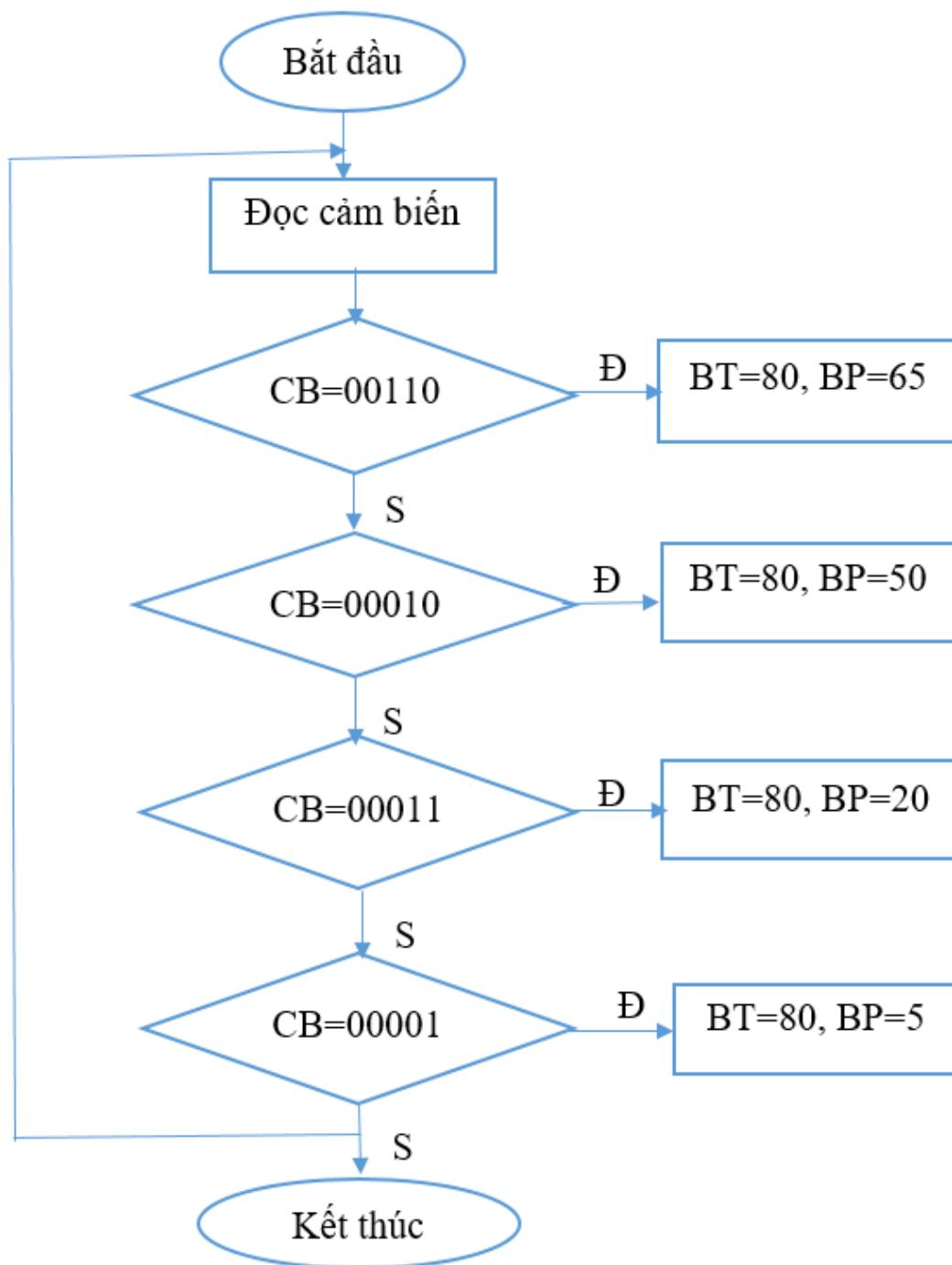
3.2.3.1 Chương trình chính:



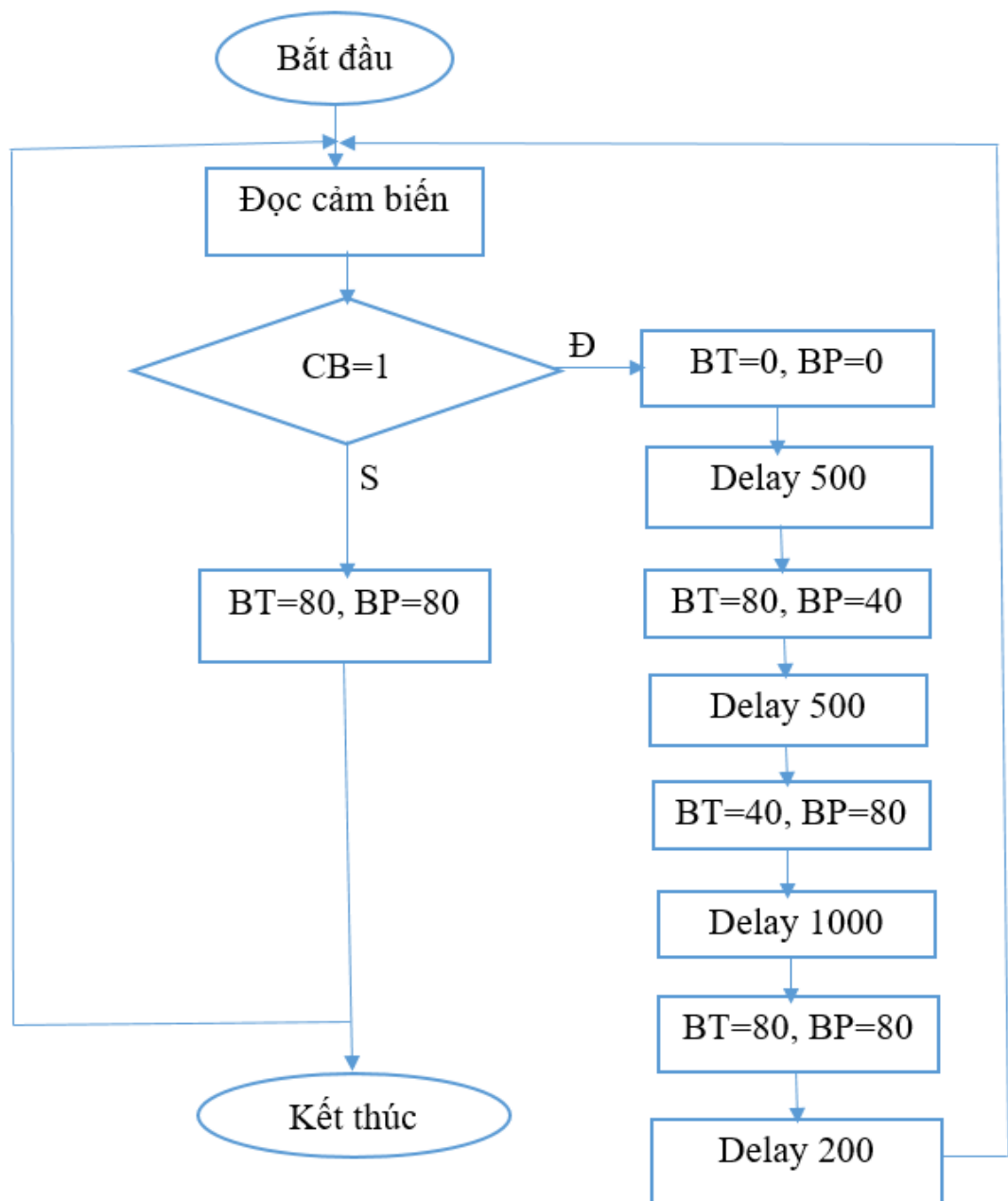
3.2.3.2 Trường hợp lệch trái:



3.2.3.3 Trường hợp lệch phải:

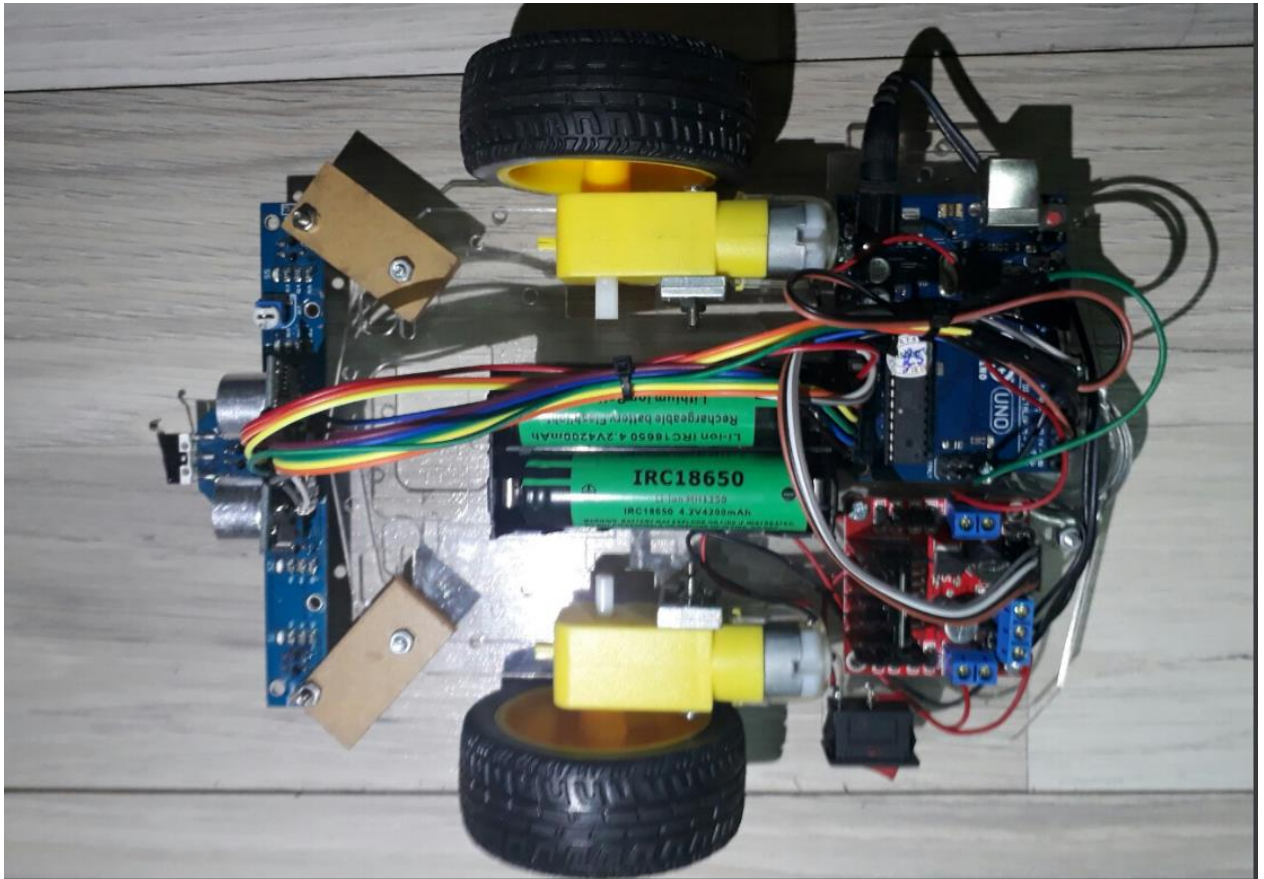


3.2.3.4 Tránh vật cản



CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

4.1. Mô hình thực tế



Hình 4.1: Mô hình thực tế

4.2. Kết quả thực nghiệm

- Thiết kế lắp ráp mô hình hoàn chỉnh.
- Thêm kinh nghiệm về lập trình arduino.
- Đạt được mục tiêu đề ra.

CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG ĐỀ TÀI

5.1. Kết quả

Sau thời gian tìm hiểu trên internet cùng với sự giúp đỡ của thầy cô và bạn bè chiếc xe của em đã hoàn thành. Chiếc xe tự động di chuyển trên đường line và tránh vật cản theo yêu cầu. Mô hình thi công và thiết kế không được thẩm mỹ cho lắm nhưng nhìn chung thì cũng đạt yêu cầu đề ra.

5.2. Hạn chế

- Vệt lố khá lớn.
- Những điểm rẽ mạnh thì xe có sai số còn lớn.
- Tốc độ xe chậm.
- Dễ bị nhiễu, hoạt động chưa ổn định.

5.3. Hướng phát triển

Sau khi hoàn thành đề tài, với những khó khăn nhất định về kiến thức chuyên ngành liên quan và giới hạn thời gian nghiên cứu hoàn thành đề tài có hạn, hệ thống điều khiển còn hạn chế một số tính năng. Để hệ thống hoàn thiện tốt hơn, có khả năng ứng dụng cao hơn trong thực tiễn, em nhận thấy cần bổ sung hoàn thiện thêm cho hệ thống một số tính năng như:

- Cải tiến xe nhanh hơn, tải trọng lớn hơn nhưng cũng phải bám đường tốt hơn.
- Xây dựng giải thuật hoàn chỉnh hơn.
- Đi lại dầy cho xe đẹp hơn.
- Thiết kế hệ thống cơ khí chính xác cho xe chạy hoàn thiện hơn .
- Lắp thêm đèn cho ô tô khi gặp vật cản đèn sáng lên.

PHỤ LỤC

1. Code chương trình

```
#define sensor PIND
#define LDIR 7
#define LPWM 6
#define RDIR 8
#define RPWM 9
#define STOP 10 // công tắc hành trình
#define detect 12 // Cảm biến phát hiện vật cản
#define trig 3 //chân trig của HC-SR05
#define echo 4//chân echo của HC-SR05
//-----
unsigned char pinmap[5]={ A0, A1, A2, A3, A4};
unsigned char temp;
unsigned char in;
unsigned char errorLeft = 0, errorRight = 1;
unsigned int dodai =0;
void setup()
{
  Serial.begin(9600);
  //-----PWM-----//
  pinMode(LDIR,OUTPUT);
  pinMode(LPWM,OUTPUT);
  pinMode(RDIR,OUTPUT);
  pinMode(RPWM,OUTPUT);
  pinMode(detect,INPUT);
  pinMode(STOP,INPUT);
```

```
digitalWrite(LDIR,LOW);
digitalWrite(RDIR,LOW);
pinMode(trig,OUTPUT);//chân trig sẽ phát tín hiệu
pinMode(echo,INPUT);//chân echo sẽ nhận tín hiệu

//-----Port Sensor-----//
for(int i=0; i<5;i++)
{
    pinMode(pinmap[i],INPUT);
}
for(int i=0; i<5;i++)
{
    digitalWrite(pinmap[i],HIGH);
}
//-----//
}

int Distance_cm ()
{
    unsigned long duration;//biến đo thời gian
    int distance;//biến lưu khoảng cách
    /* phát xung từ chân trig */
    digitalWrite(trig,0);//tắt chân trig
    delayMicroseconds(2);
    digitalWrite(trig,1);// phát xung từ chân trig
    delayMicroseconds(5);// xung có độ dài 5 microSeconds
    digitalWrite(trig,0);//tắt chân trig
    /* tính toán thời gian */
```

```
duration = pulseIn(echo,HIGH);//đo độ rộng xung HIGH ở chân echo.
distance = int(duration/2/29.412);//tính khoảng cách đến vật.

/* in kết quả ra Serial monitor */
// Serial.print(distance);
// Serial.println("cm");
// delay(200);
return distance;
}
//-----function-----//
void OnLine(int speedL,int speedR,int offset )
{
    temp= read_sensor();
    switch (temp)
    {
        case 0x04://00100
            //Robot đi thang
            speed(speedL,speedR);
            dodai=Distance_cm();
            errorLeft = 0;
            errorRight = 0;
            break;
        case 0x00:
            speed(0,0);
            dodai=Distance_cm();
            errorLeft = 0;
            errorRight = 0;
```

```
// break;

case 0x0C: //01100 Robot lech trai cap 1
    speed(speedL - 1*offset,speedR);
    dodai=Distance_cm();
    errorLeft = 1;
    errorRight = 0;
break;

case 0x08: //01000
    //Robot lech trai cap 2
    speed(speedL - 2*offset,speedR);
    dodai=Distance_cm();
    errorLeft = 1;
    errorRight = 0;
break;

case 0x18: //11000
    //Robot lech trai cap 3
    speed(speedL - 4*offset,speedR);
    dodai=Distance_cm();
    errorLeft = 1;
    errorRight = 0;
break;

case 0x10: //10000
    //Robot lech trai cap 4
    speed(speedL -5*offset,speedR);
    dodai=Distance_cm();
    errorLeft = 1;
    errorRight = 0;
```

```
        break;

//-----case lech phai-----

    case 0x06 : //00110
        //Robot lech phai cap 1
        speed(speedL,speedR - 1*offset);
        dodai=Distance_cm();
        errorLeft = 0;
        errorRight = 1;

    break;

    case 0x02 : //00010
        //Robot lech phai cap 2
        speed(speedL,speedR - 2*offset);
        dodai=Distance_cm();
        errorLeft = 0;
        errorRight = 1;

    break;

    case 0x03 : //00011
        //Robot lech phai cap 3
        speed(speedL,speedR - 4*offset);
        dodai=Distance_cm();
        errorLeft = 0;
        errorRight = 1;

    break;

    case 0x01 : //00001
        //Robot lech phai cap 4
        speed(speedL,speedR - 5*offset);
        dodai=Distance_cm();
```

```
        errorLeft = 0;
        errorRight = 1;
    break;

    //-----
    default:
        // speed(0,0);
//dodai=Distance_cm();

    break;
}
}
//-----//

//-----//
unsigned char read_sensor()
{
    char i;
    unsigned char tam=0, value=0;
    unsigned char pin[5];
    for(i=0;i<5;i++)
    {
        pin[i]=digitalRead(pinmap[i]);
    }
    for(i=0;i<5;i++)
    {
        tam=(tam|pin[i]);
    }
}
```

```
    if(i==4)
    {
        value=tam;
        break;
    }
    else tam=(tam<<1);

}
value =value;
Serial.print(value,HEX);
Serial.print("\n");
return value;
}
//-----//
void speed( int speedDC_left, int speedDC_right)
{
    if(speedDC_left>0)
    {
        analogWrite(LPWM,speedDC_left);
        digitalWrite(LDIR,LOW);
    }
    else if(speedDC_left<0)
    { speedDC_left=-speedDC_left;
      analogWrite(LPWM,speedDC_left);
      digitalWrite(LDIR,HIGH);
    }
    else if( speedDC_left==0)
```

```
{
    digitalWrite(LDIR,LOW);
    digitalWrite(LPWM,LOW);
};
if(speedDC_right>0)
{
    analogWrite(RPWM,speedDC_right);
    digitalWrite(RDIR,LOW);
}
else if(speedDC_right<0)
{ speedDC_right=-speedDC_right;
  analogWrite(RPWM,speedDC_right);
  digitalWrite(RDIR,HIGH);
}
else if (speedDC_right==0)
{
    digitalWrite(RDIR,LOW);
    digitalWrite(RPWM,LOW);
};
}

void loop()
{ while(digitalRead(STOP)==0)
  { read_sensor();
    dodai=Distance_cm();
    if(dodai>20)
      while(dodai>20)
        {
```

```
    OnLine(80,80,15  );  
}  
if(dodai<=20)  
{ while(digitalRead(STOP)==0)  
  {  
    speed(0,0);  
    delay(500);  
    speed(80,40);  
    delay(500);  
    speed(80,80);  
    delay(200);  
    speed(40,80);  
    delay(1000);  
    speed(80,80);  
    delay(200);  
    dodai=Distance_cm();  
    // if(read_sensor()==0x02||read_sensor()==0x03)  
    break;  
  }  
}  
}    speed(0,0);  
}
```

2. Giới thiệu phần mềm sử dụng

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm.

Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

Arduino IDE là phần mềm dùng để lập trình cho Arduino. Môi trường lập trình Arduino IDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh osx và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ c của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR vào chương trình nếu muốn.

3. Hướng dẫn cài đặt phần mềm

Phiên bản được viết trong bài này là phiên bản IDE 1.6.4. Đây là phiên bản mới nhất của phần mềm IDE. Sau này có thể sẽ có những phiên bản kế tiếp, tuy nhiên về cơ bản thì cách cài đặt giống nhau.

Sau khi download về thì người dùng sẽ được 1 file .exe như hình 7 bên dưới. Để cài đặt, người dùng nhấp đúp vào file đó, phần mềm sẽ bắt đầu cài đặt.

Một thông báo có thể sẽ được hiện lên với hàng chữ: “do you want to run this file?”, tức là bạn có muốn chạy file này không? Người dùng click và run (chạy) để cài chương trình.



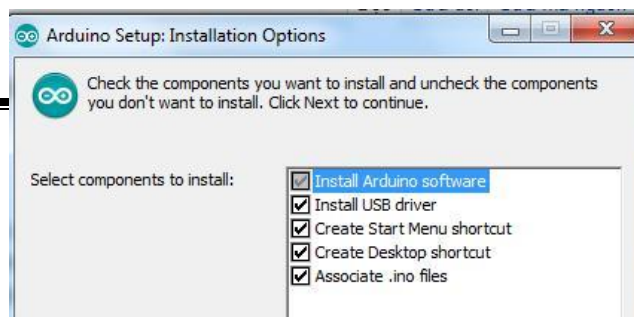
Hình 1: Bảng thông báo bắt đầu cài đặt phần mềm

Một thông báo về các điều khoản khi sử dụng phần mềm sẽ hiện ra. Người dùng nhấn vào I Agree (tôi đồng ý), để tiếp tục cài đặt chương trình.



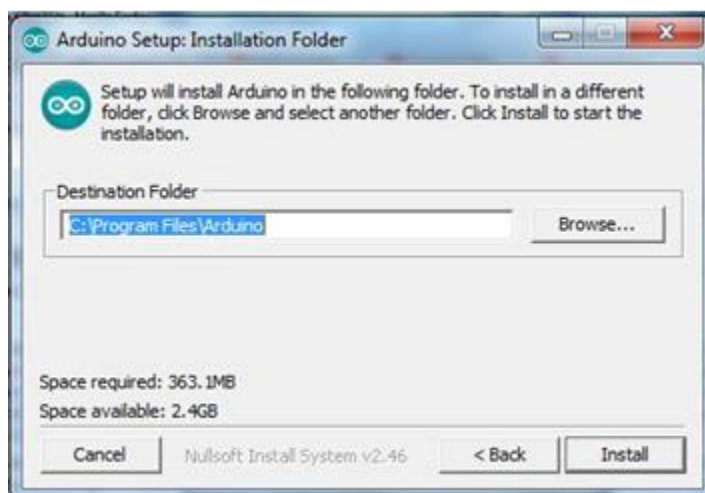
Hình 2: Thông báo về điều khoản phần mềm

Một màn hình để người dùng lựa chọn việc cài đặt các phần liên quan được hiện ra, như ở hình 10. Các bạn có thể để nguyên như vậy và bấm Next để tiếp tục quá trình cài đặt.



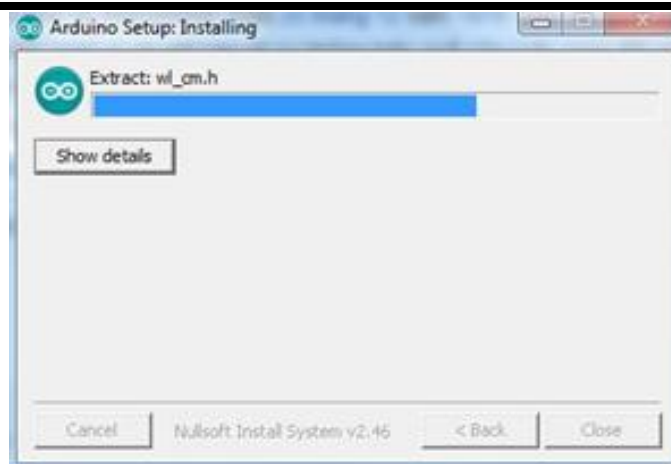
Hình 3: Các lựa chọn khi cài đặt

Phần mềm sẽ yêu cầu bạn chọn vị trí để cài đặt. Các bạn có thể để nơi cài đặt mặc định như phần mềm đưa ra, không thay đổi gì rồi bấm Install (Cài đặt).



Hình 4: Chọn đường dẫn cho phần mềm

Màn hình thể hiện việc cài đặt sẽ xuất hiện, người dùng có thể nhấp vào Show detail (Xem chi tiết) để xem chi tiết quá trình cài đặt



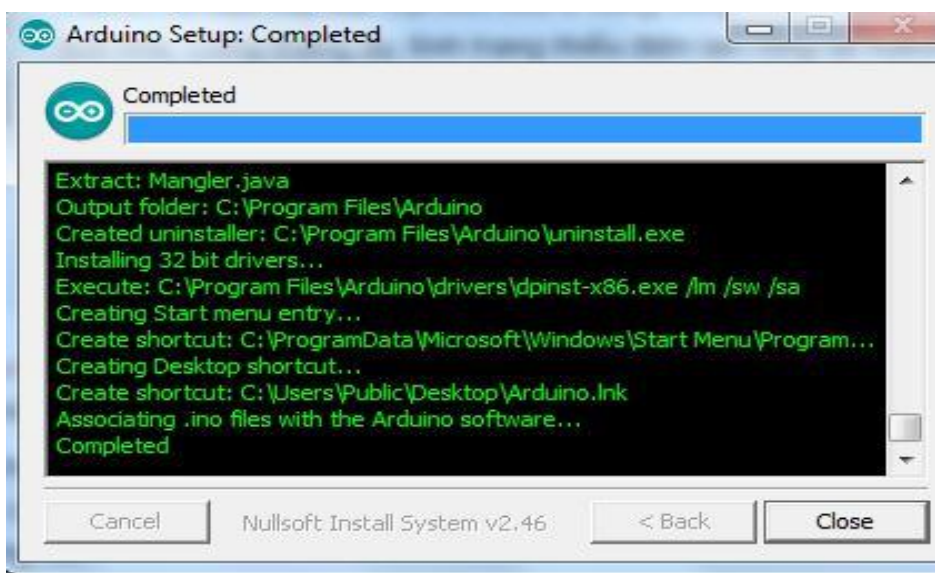
Hình 5: Quá trình cài đặt đang được thực hiện.

Trong quá trình cài đặt 1 số phiên bản sẽ hỏi có cài driver USB cho phần mềm IDE không, như hình 13 bên dưới, người dùng nên tích chọn vào ô vuông “Always trust software from “Arduino LLC””, sau đó bấm Install (Cài đặt) để cài đặt driver USB. Cần phải cài driver này thì chương trình mới nhận cổng USB của mạch Arduino



Hình 6: Cài đặt driver USB cho IDE

Sau khi việc cài đặt hoàn thành, trên màn hình sẽ xuất hiện dòng chữ Completed (Đã hoàn thành), lúc này người dùng nhấn nút Close (đóng) để hoàn tất quá trình cài đặt và bây giờ có thể sử dụng phần mềm. Biểu tượng của phần mềm sẽ xuất hiện trên màn hình Desktop và bây giờ người dùng chỉ việc nhấp đúp vào nó để sử dụng như các phần mềm khác.



Hình 7: Phần mềm đã được cài đặt xong

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Văn Hiệp, Đinh Quang Hiệp “Giáo trình lập trình Android cơ bản”, ĐH sư phạm kỹ thuật Tp. Hồ Chí Minh, năm 2015.
- [2] Trần Thế San “Cơ sở nghiên cứu và sáng tạo robot”, NXB thống kê, năm 2005.
- [3] Nguyễn Đình Phú, Trương Ngọc Anh “Giáo trình vi xử lý”, ĐH Sư phạm Kỹ thuật Tp. HCM, năm 2013.
- [4] Arduino, <http://arduino.cc>.
- [5] Dientuvietnam, <http://dientuvietnam.net>.